CLAIMS

We claim:

1. A method for using a framework module to run an application, the framework module comprising an application table and a parameter table, the application table comprising one or more application table entries, the parameter table comprising one or more parameter table entries, the method comprising:

selecting an application table entry; and

processing the selected application table entry, the processing comprising: running a global initialize function referenced by the selected application table entry, running a sub-application referenced by the selected application table entry with one or more parameters referenced by one or more parameter table entries, and running a global terminate function referenced by the selected application table entry.

2. The method of claim 1 wherein at least one of the global initialize and the global terminate functions is a NULL function.

3. The method of claim 1 further comprising:

running a module initialize function referenced by the framework module; and

running a module terminate function referenced by the framework module.

4. The method of claim 1 wherein running a sub-application comprises:

accessing from the selected application table entry a number of threads to run; and

for each of the number of threads to run, running a thread initialize function referenced by the selected application table entry, running the sub-application, and running a thread terminate function referenced by the selected application table entry.

5. The method of claim 1 further comprising:

selecting each application table entry in the application table; and

processing each selected application table entry.

6. The method of claim 1 further comprising:

collecting data specifying that a sub-application should not be run; and

wherein selecting comprises selecting an application table entry other than one that references the specified sub-application.

7. The method of claim 1 further comprising:

collecting data specifying a value of a parameter;

collecting data specifying a sub-application; and

wherein processing further comprises: if the application table entry being processed references the specified sub-application, then using the specified value of the parameter.

8. The method of claim 1 further comprising:

collecting data specifying a type of error;

collecting data specifying a sub-application;

collecting data specifying an error response action; and

wherein processing further comprises: if the application table entry being processed references the specified sub-application, and if the specified sub-application generates an error of the specified type, then performing the specified error response action.

9. The method of claim 8 wherein the error response action is in the set: break into a debugger, exit without clean up; terminate all threads; exit immediately.

10. A computer-readable medium having instructions for performing the method of claim 1.

11. A method for building a framework module for running an application, the framework module comprising an application table and a parameter table, the application comprising one or more sub-applications, the method comprising:

collecting data specifying one or more sub-applications composing the application;

collecting data specifying one or more parameters to the one or more sub-applications;

creating the application table, the creating of the application table comprising creating an application table entry for each of the one or more specified sub-applications, the creating of an application table entry comprising creating a reference to a global initialize function, creating a reference to a global terminate function, and creating a reference to the sub-application; and

creating the parameter table, the creating of the parameter table comprising creating a parameter table entry for each of the one or more specified sub-application parameters, the creating of a parameter table entry comprising creating a reference to a name of the parameter and creating a reference to a type of the parameter.

12. The method of claim 11 wherein creating a reference to at least one of the global initialize and global terminate functions comprises creating a NULL reference.

13. The method of claim 11 wherein creating an application table entry further comprises creating a reference to a number of threads to run, creating a reference to a thread initialize function, and creating a reference to a thread terminate function.

14. The method of claim 11 further comprising:

adding to the framework module a reference to a module initialize function; and

adding to the framework module a reference to a module terminate function.

15. A computer-readable medium having instructions for performing the method of claim 11.

16. A computer-readable medium having stored thereon a data structure, the data structure comprising:

a first data field containing data representing a global initialize function;

a second data field containing data representing a global terminate function; and

a third data field containing data representing an application function.

17. The data structure of claim 16 wherein the data representing at least one of the global initialize and global terminate functions are NULL data.

18. The data structure of claim 16 further comprising:

a fourth data field containing data representing an application test function.

19. The data structure of claim 16 further comprising:

a fourth data field containing data representing a number of times to call the application function.

20. The data structure of claim 19 further comprising:

a fifth data field containing data representing an application post function.

21. The data structure of claim 20 further comprising:

a sixth data field containing data representing an application post test function.

22. The data structure of claim 16 further comprising:

a fourth data field containing data representing a number of threads to run;

a fifth data field containing data representing a thread initialize function; and

a sixth data field containing data representing a thread terminate function.

23. A computer-readable medium having stored thereon a data structure, the data structure comprising:

a first data field containing data representing an application table, the application table comprising an application table entry; and

a second data field containing data representing a parameter table, the parameter table comprising a parameter table entry.

24. The data structure of claim 23 wherein the application table entry comprises:

a third data field containing data representing a global initialize function;

a fourth data field containing data representing a global terminate function; and

a fifth data field containing data representing an application function.

25. The data structure of claim 24 wherein the application table entry further comprises:

a sixth data field containing data representing an application test function.

26. The data structure of claim 24 wherein the application table entry further comprises:

a sixth data field containing data representing a number of times to call the application function.

27. The data structure of claim 26 wherein the application table entry further comprises:

a seventh data field containing data representing an application post function.

28. The data structure of claim 27 wherein the application table entry further comprises:

an eighth data field containing data representing an application post test function.

29. The data structure of claim 24 wherein the application table entry further comprises:

a sixth data field containing data representing a number of threads to run;

a seventh data field containing data representing a thread initialize function; and

an eighth data field containing data representing a thread terminate function.

30. The data structure of claim 23 wherein the parameter table entry comprises:

    a third data field containing data representing a name of a parameter;

    a fourth data field containing data representing a type of the parameter; and

    a fifth data field containing data representing a value of the parameter.

31. The data structure of claim 23 wherein the application table comprises a second application table entry.

32. The data structure of claim 23 wherein the parameter table comprises a second parameter table entry.

33. The data structure of claim 23 further comprising:

    a third data field containing data representing a module initialize function; and

    a fourth data field containing data representing a module terminate function.

34. The data structure of claim 23 further comprising:

    a third data field containing data representing a module check function; and

    a fourth data field containing data representing a module clean up function.